

BAB II

LANDASAN TEORI

2.1 Sistem Pendukung Keputusan

Sistem pendukung keputusan (Inggris: *decision support systems* disingkat DSS) adalah bagian dari sistem informasi berbasis komputer (termasuk sistem berbasis pengetahuan (manajemen pengetahuan)) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi-terstruktur yang spesifik. Menurut Moore and Chang, SPK dapat digambarkan sebagai sistem yang berkemampuan mendukung analisis *ad hoc* data, dan pemodelan keputusan, berorientasi keputusan, orientasi perencanaan masa depan, dan digunakan pada saat-saat yang tidak biasa.

Tahapan SPK:

1. Definisi masalah
2. Pengumpulan data atau elemen informasi yang relevan
3. Pengolahan data menjadi informasi baik dalam bentuk laporan grafik maupun tulisan
4. Menentukan alternatif-alternatif solusi (bisa dalam persentase)

Tujuan dari SPK:

1. Membantu menyelesaikan masalah semi-terstruktur
2. Mendukung manajer dalam mengambil keputusan suatu masalah
3. Meningkatkan efektifitas bukan efisiensi pengambilan keputusan

Dalam pemrosesannya, SPK dapat menggunakan bantuan dari sistem lain seperti *Artificial Intelligence*, *Expert Systems*, *Fuzzy Logic*, dll.

2.2 Penginapan dan Akomodasi

Penginapan atau akomodasi saat berpergian atau liburan adalah jenis tempat tinggal dalam perjalanan dimana orang yang harus tinggal jauh dari rumah lebih dari satu hari keperluan tempat untuk tidur, istirahat, keselamatan, tempat

berteduh dari suhu dingin atau hujan, penyimpanan barang, serta akses ke fungsi umum pada rumah tangga.

Penginapan dapat dilakukan pada hotel, resor, apartemen, hostel atau hostel, rumah pribadi (komersial, yaitu sebuah tempat tamu untuk tidur yang mendapatkan sarapan pagi atau rumah sewa tempat liburan, yang non-komersial dengan keanggotaan layanan keramahan atau tamu di rumah teman), dalam sebuah tenda saat berkemah (sering di perkemahan) dengan termasuk masalah sampah.

Kegiatan tidur biasanya dilakukan dengan berbaring di tempat tidur atau umumnya dengan permukaan yang lembut, seperti pada sebuah kasur, kasur udara seperti pada beberapa jenis kereta api yang menyediakan tempat tidur berbaring.

Ada kalanya penginapan dilakukan dengan tidur sambil duduk karena disebabkan tidak memungkinkan untuk berbaring misalnya dalam sebuah kereta api (atau tidur dalam mobil), bus, duduk di ruang tunggu, di jalan atau di taman, dll kursi dengan pengaturan (recline) membolehkan sesuatu antara duduk dan berbaring yang kenyamanannya tergantung pada tata letak

Agar dapat membandingkan hotel dengan jenis akomodasi yang lain, dalam bahasan berikut ini akan diuraikan secara singkat mengenai pengertian – pengertian dari jenis – jenis akomodasi sebagai berikut:

2.2.1 Hotel

Hotel adalah salah satu jenis akomodasi yang mempergunakan sebagian atau seluruh bangunan untuk menyediakan jasa pelayanan penginapan, makan dan minum serta jasa lainnya bagi umum, yang dikelola secara komersil.

2.2.2 Guest House

Sejenis akomodasi yang dapat dimiliki oleh suatu perusahaan atau instansi yang diperuntukkan bagi para tamu yang menginap dan mendapatkan pelayanan makan dan minum. pengertian aslinya, guest house merupakan akomodasi yang mempunyai fasilitas sederhana. Termasuk dalam jenis ini, di Indonesia dikenal dengan nama Pondok Wisata.

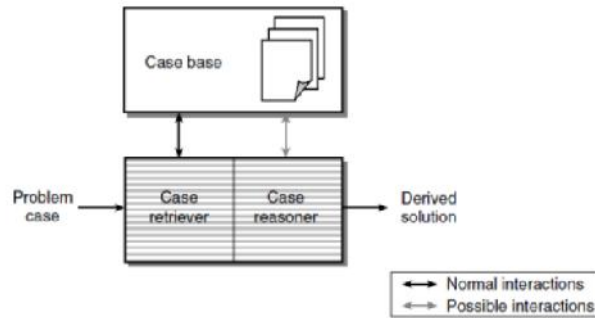
2.2.3 Homestay

adalah rumah biasa yang sebagian kamarnya disewakan kepada tamu, namun tamu yang menginap akan tinggal dalam jangka waktu lama. homestay sering diikuti oleh pelajar asing guna mempelajari kebudayaan setempat.

2.3 Metode Case-Based Reasoning

Case based reasoning adalah salah satu paradigma penyelesaian masalah dalam pendekatan Artificial Intelligence (kecerdasan buatan). Dasarnya CBR adalah metode yang digunakan untuk menyelesaikan suatu permasalahan baru dengan mengingat persamaan situasi kasus sebelumnya dan menggunakan kembali informasi serta pengetahuan pada situasi tersebut sebagai pemecahan masalahnya. CBR dapat memanfaatkan spesifikasi pengetahuan dari penelitian sebelumnya. CBR akan menyimpan pengetahuan baru yang didapat setiap kali sistem menyelesaikan masalah, sehingga pengetahuan ini dapat digunakan untuk permasalahan lain yang nantinya akan diselesaikan. Ketika masalah berhasil diselesaikan, pengetahuan yang dialami sistem disimpan agar dapat menyelesaikan masalah baru. ketika usaha untuk memecahkan masalah gagal, sistem harus mengidentifikasi dan mengingat kembali alasan mengapa terjadi kegagalan agar terhindar dari kesalahan yang sama pada kasus yang baru. Kasus terdahulu boleh diambil kembali dan dievaluasi secara berurutan ataupun bersamaan. Metode CBR mempunyai beberapa karakteristik yang membedakannya dengan metode lain. Sebuah kasus CBR biasanya diasumsikan memiliki tingkat kesempurnaan tertentu terhadap informasi yang ada didalamnya. Solusi yang diperoleh dapat digunakan pada konteks pemecahan masalah yang lainnya, sesuai dengan pendekatan informasi pada kasus-kasus tertentu (Aamodt & Plaza 2004).

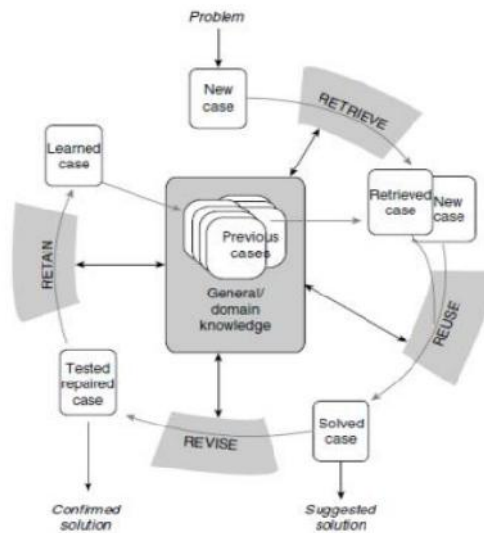
Case Based Reasoning adalah metode penyelesaian masalah dengan menggunakan solusi masalah-masalah sebelumnya yang serupa. CBR sendiri metode yang umum digunakan manusia dalam menyelesaikan sehari-hari. Layaknya metode penyelesaian masalah lainnya, Komputer dapat meniru CBR (S.K. Pal dan S.C.K. Shiu, 2004)



Gambar 2.1 Komponen CBR

Sumber (Simon C.K. Shiu dan Sankar K.Pal, 2004)

Komponen penyelesaian masalah CBR memiliki 2 bagian utama : *case retriever* dan *case reasoner*. *Case retriever* bertugas untuk mencari masalah-masalah yang mirip dengan masalah yang dimasukkan pengguna di dalam basis pengetahuan (*case base*), sementara *case reasoner* bertugas untuk mencari solusi masalah pengguna dengan melihat atau menyesuaikan solusi masalah yang ditemukan *case retriever*.



Gambar 2.2 Empat Tahapan CBR

Sumber (Simon C.K. Shiu dan Sankar K.Pal, 2004)

ada empat tahapan dalam proses CBR (Pal dan Shiu, 2004)

1. *Retrieve*, mengambil kasus-kasus lama dari *case base* yang mirip dengan kasus yang dihadapi.

2. *Reuse*, menggunakan solusi kasus-kasus lama hasil *retrieve* tersebut untuk menyelesaikan kasus yang baru tersebut.
3. *Revise*, jika diperlukan, mengadaptasi solusi kasus lama agar sesuai dengan kondisi baru
4. *Retain*, menyimpan solusi hasil *revise* yang telah divalidasi ke dalam basis data, agar dapat digunakan untuk menyelesaikan masalah serupa di masa depan.

Secara umum siklus CBR dapat di deskripsikan dengan empat proses, yaitu *retrieve*, *reuse*, *revise*, dan *retain*. *Retrieve*, pengambilan kembali kasus sebelumnya yang dapat digunakan untuk menyelesaikan kasus baru. *Reuse*, memberikan solusi terhadap kasus baru berdasarkan pengembalian kembali kasus sebelumnya. Siklus pada tahap *reuse* akan menjadi lebih sulit ketika kasus baru mempunyai perbedaan yang signifikan terhadap kasus sebelumnya. *Revise*, memperbaiki solusi yang diusulkan. *Retain*, merupakan tahap akhir dari siklus CBR, pada tahap ini sistem akan menghasilkan penyelesaian masalah terhadap kasus baru dan di gabung ke dalam sistem pengetahuan. Metode CBR sangat bergantung pada struktur dan isi dari basis kasus yang dimiliki. Selama masalah diselesaikan dengan cara memanggil kembali pengalaman sebelumnya, pencarian kasus dan proses pencocokan harus mempertimbangkan efektifitas dan efisiensi waktu. Selanjutnya, penyelesaian masalah akan disimpan kembali, informasi tersebut dapat diterapkan pada memori dan digabungkan menjadi suatu basis kasus yang baru (Kusumadewi 2009).

menyelesaikan masalah dari suatu kasus yang baru, kasus baru tersebut akan dihitung tingkat kemiripannya, kemudian tingkat kemiripan kasus tersebut akan dibandingkan dengan basis kasus yang sudah tersimpan untuk melihat penyelesaian masalah yang akan digunakan. Menghitung tingkat kemiripan suatu kasus, harus dipastikan sistem telah mempunyai basis kasus yang akan digunakan untuk mengukur tingkat kemiripan kasus, kemudian dalam suatu kasus harus memiliki variabel sebagai bahan perhitungan tingkat kemiripan (Muzid 2009). Berikut ini adalah cara menghitung tingkat kemiripan (*similarity*) pada CBR:

$$S_{total} = \frac{A1+A2+A3}{N_{Total}} \quad (2.1)$$

Penjelasan :

A1 adalah parameter harga

A2 adalah parameter lokasi

A3 adalah parameter fasilitas

N total adalah nilai total dari parameter yang di inputkan

S total adalah nilai *similarity*

Contoh Kasus :

Parameter pencarian hotel sebagai berikut:

- Harga : A1

Tabel 2.1 Parameter Harga (A1)

| Kode Harga | Harga |
|------------|--------------------------|
| 1 | Rp. 100.000 – 150.000 |
| 2 | Rp. 200.000 – 500.000 |
| 3 | Rp. 500.000 – 1.000.000 |
| 4 | Rp. 1.000.000 – 2000.000 |

- Harga : A2

Tabel 2.2 Parameter Lokasi (A2)

| Kode Lokasi | Lokasi |
|-------------|------------------------|
| 1 | Kel. Blimbing |
| 2 | Kel. Sumbersari |
| 3 | Kel. Blimbing Tlogomas |
| 4 | Kel. Gadang |
| 5 | Kel. Kotalama |

- Lokasi : A3

Tabel 2.3 Parameter Fasilitas (A3)

| Kode Fasilitas | Fasilitas |
|----------------|-------------------|
| 1 | WIFI |
| 2 | AC |
| 3 | TV |
| 4 | Kolam Renang |
| 5 | Kamar Mandi Dalam |

Parameter yang dipilih adalah A1=2, A2=4, A3=1,4

$$S_{total} = \frac{A1+A2+A3}{N \text{ Total}}$$

$$S_{total} = \frac{2+4+(F1+F4+)}{4}$$

$$S_{total} = \frac{1+1+(3)}{5} = \frac{4}{4} = 1$$

Tabel 2.4 Jumlah kemiripan dengan kasus baru

| Basis Kasus | A1 | A2 | A3 | S Total |
|-------------|----------|----------|----------|---------------------------------|
| K1 | 1 | 0 | 2 | $\frac{3}{4}$ |
| K2 | 0 | 1 | 1 | $\frac{2}{4}$ |
| K3 | 0 | 0 | 1 | $\frac{1}{4}$ |
| K4 | 1 | 0 | 1 | $\frac{2}{4}$ |
| K5 | 1 | 1 | 2 | $\frac{4}{4}$ |

Hasil perhitungan pada tabel 2.4 didapatkan satu kasus lama yang memiliki tingkat kemiripan paling tinggi dengan kasus yang baru daripada kasus-kasus lainnya, yaitu kasus K4 dengan nilai kemiripan sebesar $\frac{4}{4}$.

Tabel 2.5 Basis kasus yang tersimpan

| Basis Kasus | Kode Parameter Harga (A1) | Kode Parameter Harga (A2) | Kode Parameter Fasilitas (A3) | Solusi |
|-------------|---------------------------|---------------------------|-------------------------------|----------------------|
| K1 | (2) | (3) | (2), (4) | <i>Guest House 1</i> |
| K2 | (1) | (4) | (2) | <i>Guest House 2</i> |
| K3 | (1) | (3) | (2) | <i>Guest House 3</i> |
| K4 | (2) | (2) | (3) | <i>Guest House 4</i> |
| K5 | (2) | (4) | (1), (4) | <i>Guest House 5</i> |

Tabel 2.6 Kriteria Kemiripan

| Nilai Desimal Kemiripan | Kriteria Kemiripan |
|-------------------------|--------------------|
| 0,8 – 1 | High |
| 0,3 – 0,79 | Medium |
| 0 – 0,29 | Low |

Tabel 2.7 Hasil kriteria kemiripan setiap kasus dengan kasus baru

| Basis Kasus | S Total | Kriteria Kemiripan | Hotel |
|-------------|------------|--------------------|-------------|
| K1 | 3/4 | 0,75 | Medium |
| K2 | 2/4 | 0,5 | Medium |
| K3 | 1/4 | 0,25 | Low |
| K4 | 2/4 | 0,75 | Medium |
| K5 | 4/4 | 1 | High |

2.4 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) salah satu alat bantu dalam pengembangan sistem. UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem yang akan dibuat agar mudah dimengerti

serta melengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasi rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh *Booch Software Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan Nama *Design Object Oriented*. Metode ini menjadikan proses analisis dengan design ke dalam empat tahapan iterative., yaitu : identifikasi kelas-kelas dan obyek-obyek, identifikasi semantic dari hubungan obyek dan kelas tersebut, perincian Interface dan implementasi.

Dalam UML terdapat beberapa macam diagram yang dapat menggambarkan suatu sistem, berikut adalah beberapa diagram yang terdapat di dalam UML.

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*
4. *Class Diagram*

2.5 Responsive Web Design

Web Responsive Design merupakan salah satu teknik yang dapat membuat proses perancangan aplikasi dan situs web untuk berbagai jenis perangkat menjadi lebih mudah. Hal ini dikarenakan bahwa dengan menggunakan *responsive design*, perancangan dimungkinkan untuk dapat menerapkan solusi bagi berbagai resolusi layar, *density*, dan rasio aspek pada banyak jenis perangkat. (Syachbaa dan Zulkarnain Akib, 2014)

Web Responsive design pertama kali diperkenalkan oleh Ethan Marcotte pada artikelnya yang inspiratif yaitu *Web design Responsive*. Semakin Banyaknya perangkat dengan berbagai ukuran yang berbeda, maka situs perlu untuk mengenali ukuran perangkat pengguna. (Agus, 2014)

2.6 PHP

PHP singkatan dari *PHP Hypertext Preprocessor* yang digunakan sebagai bahasa *scriptserver-side* dalam pengembangan web yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan web dapat dibuat dinamis

sehingga *maintenance* situs web tersebut menjadi lebih mudah dan efisien. PHP merupakan *software* atau perangkat lunak yang bersifat *Open-Source* yang disebar dan dilisensikan secara gratis serta dapat di *download* secara bebas. PHP ditulis menggunakan bahasa C.

PHP diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. Awalnya, PHP digunakan untuk mencatat jumlah serta untuk mengetahui siapa saja pengunjung pada *homepage*-nya. Rasmus Lerdorf adalah salah seorang pendukung *Open-Source*. PHP mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis, kemudian menambah kemampuan PHP 1.0 dan meluncurkan PHP 2.0. Tahun 1996, PHP telah banyak digunakan dalam website di dunia. Sebuah kelompok pengembang *software* yang terdiri dari Rasmus, Zeew Suraski, Andi Gutman, Stig Bakken, Shane Caraveo, dan Jum Winstead bekerja sama untuk menyempurnakan PHP 2.0. Akhirnya, pada tahun 1998, PHP 3.0 diluncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan PHP 4.0. Tidak berhenti sampai disitu, kemampuan PHP terus ditambah.

2.7 JavaScript

JavaScript adalah bahasa pemrograman yang ditafsirkan bahwa sebagian besar digunakan untuk mengubah halaman web statis menjadi halaman dinamis dan interaktif setelah web browser telah selesai men-download sebuah halaman web. Misalnya, jam ditampilkan pada halaman yang update sendiri untuk menunjukkan waktu saat ini pada komputer pengguna. Desain JavaScript dipengaruhi oleh banyak bahasa pemrograman, termasuk C, tetapi dimaksudkan untuk lebih digunakan oleh *non-programmer*. JavaScript seringkali disertakan dalam file HTML atau link dari file HTML dan dijalankan secara lokal oleh web browser. Berarti bahwa *server* bebas untuk mengerjakan sesuatu yang lain daripada pemrosesan instruksi untuk setiap klien. Hal ini telah membuat JavaScript pilihan yang lebih populer daripada bahasa yang memerlukan *server* untuk melakukan pengolahan.

JavaScript pertama kali dikembangkan oleh Brendan Eich dari Netscape dibawah nama Mocha, yang nantinya namanya diganti menjadi LiveScript, dan

akhirnya menjadi JavaScript. Navigator sebelumnya telah mendukung Java untuk lebih bisa dimanfaatkan para *programmer* yang non-Java. Maka dikembangkanlah bahasa pemrograman bernama LiveScript untuk mengakomodasi hal tersebut. Bahasa pemrograman inilah yang akhirnya berkembang dan diberi nama JavaScript, walaupun tidak ada hubungan bahasa antara Java dengan JavaScript.

JavaScript bisa digunakan untuk banyak tujuan, misalnya untuk membuat efek *rollover* baik di gambar maupun teks, dan yang penting juga adalah untuk membuat AJAX. JavaScript adalah bahasa yang digunakan untuk AJAX.

2.8 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (DBMS) yang *multi-thread*, dan *multi-user*. MySQL adalah implementasi dari sistem manajemen basis data relasional (RDBMS). MySQL dibuat oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah *database* berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris.

MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL. Pendiri MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael "Monty". Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL (*General Public License*) namun tidak boleh menjadikan produk turunan yang bersifat komersial.

Saat ini MySQL merupakan *databaseserver* yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses *database* yaitu SQL. SQL (*Structured Query Language*) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama sistem R. Kemudian SQL juga dikembangkan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan *database* lebih *user-friendly* dibandingkan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni.

SQL dapat digunakan secara berdiri sendiri maupun di lekatkan pada bahasa pemrograman seperti Python, Ruby, PHP, C++, dan lainnya.

2.9 Codeigniter

Codeigniter adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibandingkan jika menulis semua code ke program dari awal. (Koespradono dkk, 2013)

Codeigniter menyediakan banyak *library* untuk mengerjakan tugas tugas yang umumnya ada pada sebuah aplikasi berbasis *web*. Selain itu struktur dan susunan logis dari *Codeigniter* membuat palikasi yang dibuat menjadi semakin teratur dan rapi. *Codeigniter* dibangun menggunakan konsep Model-View-Controller development pattern. *Codeigniter* sendiri merupakan salah satu framework tercepat dibandingkan framework lainnya.

Kelebihan *Codeigniter* :

1. Menggunakan Pattern MVC
Dengan menggunakan pattern MVC ini, struktur kkode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas
2. URL Friendly
URL yang dihasilkan sangat friendly, diminimalisasi penggunaan \$GET dan digantikan dengan URL.
3. Kemudahan
Kemudahan dalam mempelajari, membuat library dan helper, memodifikasi serta mngintegrasikan library dan helper.

Jika kita mmbandingkan antara codeigniter dengan framework lainnya maka beberapa poin yang membuat codeigniter unggul adalah :

1. Kecepatan
Berdasarkan hasil benchmark codeigniter merupakan salah satu framework PHP tercepat yang ada saat ini.
2. Mudah dimodifikasi dan beradaptasi
Sangat mudah memodifikasi *behavior framework* ini. Tidak membutuhkan *server requirement* yang macam-macam serta mudah mengadopsi *library* lainnya.
3. Dokumen lengkap dan jelas

Mudah mendapatkan informasi tentang *codeigniter* melalui buku dan referensi lainnya

4. *Learning curve* rendah

Codeigniter sangat mudah dipelajari. Pemilihan *framework* hal ini sangat penting diperhatikan karena kita juga juga harus memperhatikan *skill* individu. Jika *framework* sangat sulit dipelajari maka akan beresiko untuk memperlambat proses *development*.

